

FILEID**LASWEEP

B 10

LPA
V04

LL	AAAAAA	SSSSSSSS	WW	WW	EEEEEEEEE	EEEEEEEEE	PPPPPPP
LL	AAAAAA	SSSSSSSS	WW	WW	EE	EE	PPPPPPP
LL	AA	AA	SS	WW	WW	EE	PP
LL	AA	AA	SS	WW	WW	EE	PP
LL	AA	AA	SS	WW	WW	EE	PP
LL	AA	AA	SS	WW	WW	EE	PP
LL	AA	AA	SSSSSS	WW	WW	EEEEEEE	PPPPPPP
LL	AA	AA	SSSSSS	WW	WW	EEEEEEE	PPPPPPP
LL	AA	AA	SSSSSS	WW	WW	EE	PP
LL	AA	AA	SSSSSS	WW	WW	EE	PP
LL	AA	AA	SSSSSS	WW	WW	EE	PP
LL	AA	AA	SSSSSS	WW	WW	EE	PP
LL	AA	AA	SSSSSS	WW	WW	EE	PP
LLLLLLLLL	AA	AA	SSSSSSSS	WW	WW	EEEEEEEEE	PP
LLLLLLLLL	AA	AA	SSSSSSSS	WW	WW	EEEEEEEEE	PP

LL		SSSSSSSS
LL		SSSSSSSS
LL		SS
LL		SS
LL		SS
LL		SSSSSS
LL		SSSSSS
LL		SS
LLLLLLLLL		SSSSSSSS
LLLLLLLLL		SSSSSSSS

(2)	53	DECLARATIONS
(3)	77	LPA\$SETIBF - INITIALIZE IBUF ARRAY
(4)	175	START SWEEP ROUTINES
(5)	433	LPA\$STPSWP - STOP SWEEP
(6)	502	LPA\$CLOCKA - SET CLOCK A RATE
(7)	562	LPA\$CLOCKB - SET CLOCK B RATE
(8)	673	LPA\$LAMSKS - SET MASKS BUFFER
(9)	778	LPA\$SETADC - SET CHANNEL PARAMETERS
(10)	875	LPA\$CVADF - CONVERT A/D TO FLOATING POINT
(10)	876	LPA\$FLT16 - CONVERT UNSIGNED WORD TO FLOATING POINT
(11)	942	LPA\$XRATE - COMPUTE CLOCK RATE AND PRESET
(12)	1051	LPA\$LOADMC - LOAD MICROCODE
(13)	1148	LPA\$ASSIGN - ASSIGN A CHANNEL TO AN LPA-11

0000 1 :TITLE LPASSWEEP
0000 2 :IDENT 'V04-000'
0000 3 *****
0000 4 *
0000 5 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 6 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 7 * ALL RIGHTS RESERVED.
0000 8 *
0000 9 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 10 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 11 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 12 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 13 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 14 * TRANSFERRED.
0000 15 *
0000 16 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 17 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 18 * CORPORATION.
0000 19 *
0000 20 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 21 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 22 *
0000 23 *
0000 24 *
0000 25 *
0000 26 *****
0000 27 :
0000 28 :
0000 29 :++
0000 30 :FACILITY: LPA-11 PROCEDURE LIBRARY
0000 31 :
0000 32 :ABSTRACT:
0000 33 : THIS MODULE CONTAINS THE START SWEEP ROUTINES FOR THE LPA-11
0000 34 : PROCEDURE LIBRARY.
0000 35 :
0000 36 :ENVIRONMENT: USER MODE, SHARED OR NON-SHARED LIBRARY
0000 37 :
0000 38 :AUTHOR: STEVE BECKHARDT. CREATION DATE: 23-AUG-78
0000 39 :
0000 40 :MODIFIED BY:
0000 41 :
0000 42 : V03-001 SBL3001 Steven B. Lionel 30-Mar-1982
0000 43 : Change module name to LPASSWEEP.
0000 44 :
0000 45 : V04 SRB0001 STEVE BECKHARDT 23-OCT-1979
0000 46 : FIXED BUG IN HANDLING OF BUFFER OVERRUN NON-FATAL:
0000 47 : INITIALIZED BUFFER 0 RELEASED FLAG TO 1 IN ROUTINE
0000 48 : SWPCOM IF BUFFER OVERRUN IS NON-FATAL. AS A RESULT,
0000 49 : BUFFER 0 SHOULD NOT BE RELEASED THE FIRST TIME.
0000 50 :
0000 51 :--

DECLARATIONS

0000 53 .SBTTL DECLARATIONS
0000 54 : INCLUDE FILES:
0000 55 :
0000 56 :
0000 57 :
0000 58 : MACROS:
0000 59 :
0000 60 :
0000 61 :
0000 62 :
0000 63 : EQUATED SYMBOLS:
0000 64 :
0000 65 :
0000 66 :
0000 67 : OWN STORAGE:
0000 68 :
0000 69 :
00000000 70 .PSECT _LPA\$CODE,PIC,SHR,EXE,NOWRT,BYTE
0000 71 :
0000 72 : PREFIX STRING USED IN ASSIGNING CHANNEL
0000 73 :
24 31 31 41 50 4C 0000 74 DNPREFIX: ASCII /LPA11\$/
00000006 0006 75 DNPREFIXS=-DNPREFIX

0006 77 .SBTTL LPASSETIBF - INITIALIZE IBUF ARRAY
 0006 78 :++
 0006 79 : FUNCTIONAL DESCRIPTION:
 0006 80 :
 0006 81 : THIS ROUTINE INITIALIZES THE IBUF ARRAY
 0006 82 :
 0006 83 : CALLING SEQUENCE:
 0006 84 :
 0006 85 : CALLS/G
 0006 86 :
 0006 87 : INPUT PARAMETERS:
 0006 88 :
 0006 89 : IBUF(AP) ADDRESS OF IBUF ARRAY (MUST BE LONGWORD ALIGNED)
 0006 90 : IND(AP) ADDRESS OF LONGWORD TO RECEIVE COMPLETION STATUS
 0006 91 : LAMSKB(AP) ADDRESS OF LAMSKS ARRAY
 0006 92 : BUFO(AP) ADDRESS OF FIRST DATA BUFFER
 0006 93 :
 0006 94 :
 0006 95 : BUFN(AP) ADDRESS OF LAST DATA BUFFER (UP TO 8 ARE ALLOWED)
 0006 96 :
 0006 97 : IMPLICIT INPUTS:
 0006 98 :
 0006 99 : NONE
 0006 100 :
 0006 101 : OUTPUT PARAMETERS:
 0006 102 :
 0006 103 : IND(AP) : ADDRESS OF LONGWORD TO RECEIVE COMPLETION STATUS
 0006 104 :
 0006 105 : IMPLICIT OUTPUTS:
 0006 106 :
 0006 107 : NONE
 0006 108 :
 0006 109 : COMPLETION CODES:
 0006 110 :
 0006 111 : 1 INDICATES IBUF WAS SUCCESSFULLY INITIALIZED
 0006 112 : 0 INDICATES AN ARGUMENT ERROR - POSSIBLE CAUSES:
 0006 113 : 1) INCORRECT NUMBER OF ARGUMENTS
 0006 114 : 2) IBUF ARRAY NOT LONGWORD ALIGNED
 0006 115 : 3) BUFFER ADDRESSES NOT EQUIDISTANT
 0006 116 :
 0006 117 : SIDE EFFECTS:
 0006 118 :
 0006 119 : NONE
 0006 120 :
 0006 121 :--
 0006 122 :
 007C 0006 123 :.ENTRY LPASSETIBF,"M<R2,R3,R4,R5,R6>
 0008 124 :
 56 04 52 D4 0008 125 : CLRL R2 : RETURN STATUS
 56 03 D3 000E 126 : MOVL 4(AP),R6 : GET ADDRESS OF IBUF ARRAY
 6F 12 0011 127 : BITL #3,R6 : IS IBUF LONGWORD ALIGNED?
 54 66 00 2C 0013 128 : BNEQ 60\$: NO - ERROR
 54 0C AC DE 001B 129 : MOVC5 #0,(R6),#0,#IBFSK_LENGTH,(R6) : ZERO IBUF
 001F 130 : MOVAL 12(AP),R4 : POINT TO LAMSKB ARG
 1C A6 84 D0 001F 131 :
 132 : PROCESS LAMSKB ARG
 133 : MOVL (R4)+,IBFSL_LAMSKB(R6) ; STORE IN IBUF

66 00A8 BF 00	52 D4 0008 125 : CLRL R2 : RETURN STATUS
	56 03 D3 000E 126 : MOVL 4(AP),R6 : GET ADDRESS OF IBUF ARRAY
	6F 12 0011 127 : BITL #3,R6 : IS IBUF LONGWORD ALIGNED?
	54 66 00 2C 0013 128 : BNEQ 60\$: NO - ERROR
	54 0C AC DE 001B 129 : MOVC5 #0,(R6),#0,#IBFSK_LENGTH,(R6) : ZERO IBUF
	001F 130 : MOVAL 12(AP),R4 : POINT TO LAMSKB ARG
	1C A6 84 D0 001F 131 : 132 : PROCESS LAMSKB ARG 133 : MOVL (R4)+,IBFSL_LAMSKB(R6) ; STORE IN IBUF

LPASSETIBF - INITIALIZE IBUF ARRAY

			0023	134		
			0023	135		
			0023	135		
53	6C	9A	0023	136	MOVZBL (AP), R3	: PROCESS BUFFER ADDRESSES
53	04	C2	0026	137	SUBL #4, R3	: GET NUMBER OF ARGS
07	53	D1	0029	138	CMPL R3, #7	: ACCOUNT FOR FIRST ARGUMENTS
	54	1A	002C	139	BGTRU 60\$: R3 = # OF BUFFERS - 1
2C	A6	64	D0	140	MOVL (R4), IBFSL_CMDTBL+CMTSL_BFRADDR(R6); STORE FIRST BUFFER ADDR.	: INCORRECT # OF ARGS
22	A6	53	90	141	MOVB R3, IBFSL_CMDTBL+CMTSB_VBFRMASK(R6)	: STORE # IN COMMAND TABLE
14	A6	14	13	142	BEQL 50\$: ONLY 1 BUFFER ADDRESS SPECIFIED
14	A6	64	84	143	SUBL3 (R4)+, (R4), IBFSL_LBUF(R6)	: COMPUTE AND STORE BUFFER LENGTH
		0A	11	144	BRB 40\$	
50	64	84	C3	003F	145	
14	A6	50	D1	0043	30\$: SUBL3 (R4)+, (R4), R0	: COMPUTE NEXT LENGTH
		39	12	0047	147 CMPL R0, IBFSL_LBUF(R6)	: MAKE SURE IT AGREES
	F3	53	F5	0049	148 BNEQ 60\$: IT DOESN'T - ERROR
				004C	149 40\$: SOBGTR R3, 30\$: DO NEXT ONE
				004C	150	
50	A6	50	A6	DE	004C 151 50\$: MOVAL IBFSL_USRQFL(R6), IBFSL_USRQFL(R6)	: INITIALIZE ARGUMENT INDEPENDENT STUFF. FIRST INIT. QUEUES
54	A6	50	A6	DE	0051 152 MOVAL IBFSL_USRQFL(R6), IBFSL_USRQBL(R6)	: USER QUEUE
58	A6	58	A6	DE	0056 153 MOVAL IBFSL_DEVQFL(R6), IBFSL_DEVQFL(R6)	: DEVICE QUEUE
5C	A6	58	A6	DE	005B 154 MOVAL IBFSL_DEVQFL(R6), IBFSL_DEVQBL(R6)	
60	A6	60	A6	DE	0060 155 MOVAL IBFSL_INUQFL(R6), IBFSL_INUQFL(R6)	: IN USE QUEUE
64	A6	60	A6	DE	0065 156 MOVAL IBFSL_INUQFL(R6), IBFSL_INUQBL(R6)	
24	A6	48	A6	3E	006A 157 MOVAL IBFSL_USW(R6), IBFSL_CMDTBL+CMTSL_USWADDR(R6)	: USW ADDRESS
38	A6	00010001	01000001	BF	7D 006F 158 MOVQ #^X1000101000001, IBFSL_CMDTBL+CMTSW_DELAY(R6); DEFAULT SAMPLING	
				007B 159		: NUMBERS
				007B 160		
66	1234	8F	B0	007B	161 MOVW #INITCODE, IBFSQ_IOST(R6)	: SHOW THAT SETIBF WAS CALLED
				0080 162		
				0080 163		
				0082 164	INCL R2	: INDICATE SUCCESS
51	08	AC	D0	0082	170 60\$: MOVL 8(AP), R1	: GET ADDRESS OF IND
	03	13	0086	171	BEQL 70\$: DEFULTED
61	52	D0	0088	172	MOVL R2, (R1)	: RETURN STATUS
		04	008B	173 70\$: RET		

START SWEEP ROUTINES

008C 175 .SBTTL START SWEEP ROUTINES
 008C 176 ++
 008C 177 FUNCTIONAL DESCRIPTION:
 008C 178 THESE ROUTINES ARE THE START SWEEP ROUTINES. THERE ARE
 008C 180 FOUR TYPES OF SWEEPS: A/D, D/A, DIGITAL IN, AND DIGITAL OUT.
 008C 181
 008C 182 CALLING SEQUENCE:
 008C 183
 008C 184 CALLS/G
 008C 185
 008C 186 INPUT PARAMETERS:
 008C 187
 008C 188 IBUF(AP) ADDRESS OF ARRAY INITIALIZED BY SETIBF
 008C 189 LBUF(AP) ADDRESS OF WORD CONTAINING THE SIZE OF EACH DATA BUFFER
 (IN WORDS)
 008C 190
 008C 191 NBUF(AP) ADDRESS OF LONGWORD CONTAINING NUMBER OF BUFFERS TO FILL
 008C 192 MODE(AP) ADDRESS OF A WORD WHICH SPECIFIES SAMPLING OPTIONS
 008C 193 BIT 5 SERIAL/PARALLEL (DUAL A/D)
 008C 194 BIT 6 DEDICATED/MULTIREQUEST MODE
 008C 195 BIT 9 CLOCK OVERFLOW/EXTERNAL TRIGGER
 008C 196 BIT 10 TIME STAMPING
 008C 197 BIT 11 EVENT MARKING
 008C 198 BIT 12 IMMEDIATE START/DIGITAL INPUT START
 008C 199 BIT 13 SINGLE/DUAL A/D
 008C 200 BIT 14 BUFFER OVER/UNDERRUN FATAL/NON-FATAL
 008C 201 DWELL(AP) ADDRESS OF WORD CONTAINING DWELL VALUE
 008C 202 IEFN(AP) VALUE OF EVENT FLAG OR IF GREATER THAN 128 ADDRESS OF
 008C 203 COMPLETION ROUTINE
 008C 204 LDELAY(AP) ADDRESS OF A WORD CONTAINING DELAY VALUE
 008C 205 ICHN(AP) ADDRESS OF A BYTE CONTAINING START CHANNEL VALUE
 008C 206 NCHN(AP) ADDRESS OF A WORD CONTAINING NUMBER OF SAMPLES VALUE
 008C 207 IND(AP) ADDRESS OF A LONGWORD TO RECEIVE STATUS
 008C 208
 008C 209 IMPLICIT INPUTS:
 008C 210
 008C 211 VARIOUS FIELDS IN THE IBUF ARRAY
 008C 212
 008C 213 OUTPUT PARAMETERS:
 008C 214
 008C 215 IND(AP) ADDRESS OF A LONGWORD TO RECEIVE STATUS
 008C 216
 008C 217 IMPLICIT OUTPUTS:
 008C 218
 008C 219 VARIOUS FIELDS IN THE IBUF ARRAY
 008C 220
 008C 221 COMPLETION CODES:
 008C 222
 008C 223 1 INDICATES SUCCESS
 008C 224 0 INDICATES ERROR DETECTED BY THIS ROUTINE - POSSIBILITIES:
 008C 225 1) SETIBF WAS NOT CALLED FIRST
 008C 226 2) RLSBUF HAS NOT BEEN CALLED TO RELEASE A BUFFER
 008C 227 3) SIZE OF DATA BUFFERS DISAGREES WITH SIZE
 008C 228 COMPUTED FROM SETIBF CALL.
 008C 229 VARIOUS VMS CODES RETURNED BY \$ASSIGN AND \$QIO
 008C 230
 008C 231 SIDE EFFECTS:

START SWEEP ROUTINES

```

008C 232 : VARIOUS FIELDS IN THE IBUF ARRAY ARE MODIFIED
008C 233 :
008C 234 :
008C 235 :--+
008C 236
55 02 00FC 008C 237 .ENTRY LPA$ADSWP,^M<R2,R3,R4,R5,R6,R7>
17 D0 008E 238 MOVL #2,R5 ; MODE WORD ; START A/D SWEEP
11 0091 239 BRB SWPCOM

55 0082 00FC 0093 242 .ENTRY LPA$DASWP,^M<R2,R3,R4,R5,R6,R7>
8F 3C 0095 243 MOVZWL #^X82,R5 ; MODE WORD ; START D/A SWEEP
OE 11 009A 244 BRB SWPCOM

55 1A 00FC 009C 247 .ENTRY LPA$DISWP,^M<R2,R3,R4,R5,R6,R7>
07 D0 009E 248 MOVL #^X1A,R5 ; MODE WORD ; START DIG. INPUT SWEEP
11 00A1 249 BRB SWPCOM

55 009A 00FC 00A3 252 .ENTRY LPA$DOSWP,^M<R2,R3,R4,R5,R6,R7>
8F 3C 00A5 253 MOVZWL #^X9A,R5 ; START DIG. OUT SWEEP
00AA 254
00AA 255
00AA 256

57 04 AC DE 00AA 257 SWPCOM: ; COMMON PROCESSING FOR ALL SWEEP ROUTINES. R5 CONTAINS MODE WORD.
53 6C 02 83 00AA 258 MOVAL 4(AP),R7 ; POINT TO FIRST ARGUMENT
00AE 259 SUBB3 #2,(AP),R3 ; R3 CONTAINS # OF OPTIONAL ARGS
00B2 260
50 D4 00B2 261 CLRL R0 ; ASSUME ERROR
00B4 262
1234 56 87 D0 00B4 263 MOVL (R7)+,R6 ; ADDRESS OF IBUF ARRAY
8F 66 B1 00B7 264 CMPW IBFSQ_IOST(R6),#INITCODE ; VERIFY SETIBF WAS CALLED
18 12 00BC 265 BNEQ 15$ ; IT WASN'T - ERROR
66 7C 00BE 266 CLRQ IBFSQ_IOST(R6) ; CLEAR USER'S I/O STATUS BLOCK
00C0 267
11 4C A6 00 E1 00C0 268 ; VERIFY THAT A BUFFER HAS BEEN RELEASED AND SET IN USW
00C0 269 BBC #FLG_V_USWSET,IBFSW_FLAGS(R6),15$ ; BR. IF USW IS NOT SET
00C5 270
00C5 271 ; PROCESS LBUF
51 97 3C 00C5 272 MOVZWL @R7+,R1 ; GET LBUF
51 02 C4 00C8 273 MULL #2,R1 ; MULTIPLY BY 2 TO CONV. WORDS TO BYTES
52 14 A6 D0 00CB 274 MOVL IBFSL_LBUF(R6),R2 ; GET LBUF CALCULATED IN SETIBF
08 13 00CF 275 BEQL 20$ ; THERE ISN'T ONE
52 51 D1 00D1 276 CMPL R1,R2 ; COMPARE THEM
03 13 00D4 277 BEQL 20$ ; EQUAL - NO ERROR
0160 31 00D6 278
00D9 279 15$: BRW 115$ ; ERROR
14 A6 51 D0 00D9 280
52 22 A6 9A 00DD 281 20$: MOVL R1,IBFSL_LBUF(R6) ; STORE LENGTH OF EACH BUFFER
52 52 D6 00E1 282 MOVZBL IBFSL_CMDTBL+CMTSB_VBFMRMASK(R6),R2 ; GET # OF BUFFERS-1
00E3 283 INCL R2 ; ADD 1
28 A6 51 52 C5 00E8 284 MULL3 R2,R1,IBFSL_CMDTBL+CMTSL_BFRLEN(R6) ; STORE OVERALL BFR LENGTH
00E8 285
00E8 286 ; PROCESS NBUF
53 97 00E8 287 DECB R3 ; DECR. ARG COUNT
0F 19 00EA 288 BLSS 30$ ; ARG OMITTED

```

START SWEEP ROUTINES

```

51 87 D0 00EC 289      MOVL (R7)+,R1          ; GET ADDRESS OF NBUF
0A 13 00EF 290      BEQL 30$                 ; DEFAULTED (CONTINUOUS SAMPLING)
18 A6 61 D0 00F1 291      MOVL (R1),IBFSL_NBUF(R6) ; STORE NBUF
04 13 00F5 292      BEQL 30$                 ; ZERO ALSO MEANS CONTINUOUS SAMPLING
4C A6 04 AB 00F7 293      BISW #FLG_M_CNTBFRS,IBFSW_FLAGS(R6) ; SET COUNT BUFFERS FLAG
                           00FB 294
                           00FB 295 30$: ; PROCESS MODE
                           53 97 00FB 296      DECB R3              ; DECR. ARG COUNT
                           4C 19 00FD 297      BLSS 50$             ; ARG. OMITTED
                           51 87 D0 00FF 298      MOVL (R7)+,R1          ; GET ADDRESS OF MODE
                           47 13 0102 299      BEQL 50$             ; DEFAULTED
                           51 61 3C 0104 300      MOVZWL (R1),R1        ; GET MODE
                           0107 301
                           0107 302 ; SET BITS IN MODE WORD (IN R5) DEPENDING ON BITS IN MODE ARG (IN R1)
04 51 05 E1 0107 303      BBC #5,R1,32$         ; BRANCH IF SERIAL
00 55 0B E2 010B 304      BBSS #11,R5,32$       ; SET FOR PARALLEL
04 51 06 E1 010F 305 32$: BBC #6,R1,34$         ; BRANCH IF DEDICATED MODE
00 55 03 E2 0113 306      BBSS #3,R5,34$         ; SET FOR MULTIREQUEST MODE
04 51 09 E1 0117 307 34$: BBC #9,R1,36$         ; BRANCH IF CLOCK OVERFLOW TRIGGER
00 55 0A E2 011B 308      BBSS #10,R5,36$        ; SET FOR EXTERNAL TRIGGER
04 51 0A E1 011F 309 36$: BBC #10,R1,38$        ; BRANCH IF NO TIME STAMPING
00 55 0F E2 0123 310      BBSS #15,R5,38$        ; SET FOR TIME STAMPING
04 51 0B E1 0127 311 38$: BBC #11,R1,40$        ; BRANCH IF NO EVENT MARKING
00 55 0E E2 012B 312      BBSS #14,R5,40$        ; SET FOR EVENT MARKING
04 51 0C E1 012F 313 40$: BBC #12,R1,42$        ; BRANCH IF IMMEDIATE START
00 55 0C E2 0133 314      BBSS #12,R5,42$        ; SET FOR DIGITAL INPUT START
04 51 0D E1 0137 315 42$: BBC #13,R1,44$        ; BRANCH IF SINGLE A/D CONVERTER
00 55 05 E2 013B 316      BBSS #5,R5,44$         ; SET FOR DUAL A/D CONVERTERS
08 51 0E E1 013F 317 44$: BBC #14,R1,50$        ; BRANCH IF OVER/UNDERRUN IS FATAL
4C A6 10 AB 0143 318      BISW #FLG_M_BFR0RLSD,IBFSW_FLAGS(R6) ; SET BUFFER 0 RELEASED FLAG
00 55 17 E2 0147 319      BBSS #23,R5,50$        ; SET FOR OVER/UNDERRUN NON-FATAL
                           014B 320
                           014B 321 50$: ; MODE WORD NOW COMPLETE EXCEPT FOR CHANNEL SELECTION BITS (BITS 8-9)
20 A6 55 C8 014B 322      BISL R5,IBFSL_CMDTBL+CMTSW_MODE(R6) ; OR INTO MODE WORD IN CMD TBL
                           014F 323
                           014F 324 ; PROCESS DWELL
                           53 97 014F 325      DECB R3              ; DECR. ARG COUNT
                           09 19 0151 326      BLSS 60$             ; ARG OMITTED
                           51 87 D0 0153 327      MOVL (R7)+,R1          ; ADDRESS OF DWELL VALUE
                           04 13 0156 328      BEQL 60$             ; DEFAULTED
                           3E A6 61 B0 0158 329      MOVW (R1),IBFSL_CMDTBL+CMTSW_DWELL(R6) ; STORE DWELL
                           015C 330
                           015C 331 60$: ; PROCESS IEFN (EVENT FLAG OR COMPLETION ROUTINE ADDRESS)
                           54 0000*8F 3C 015C 332      MOVZWL #IOS_STARTDATA!IOSM_SETEVF,R4 ; I/O FUNCTION CODE
                           53 97 0161 333      DECB R3              ; DECR. ARG COUNT
                           17 19 0163 334      BLSS 65$             ; ARG OMITTED
                           50 87 D0 0165 335      MOVL (R7)+,R0          ; GET EVENT FLAG # OR ADDRESS OF ROUTINE
                           12 13 0168 336      BEQL 65$             ; EITHER DEFAULTED OR EVENT FLAG ZERO
                           00000080 8F 50 D1 016A 337      CMPL R0,#128        ; EVENT FLAG OR AST ADDRESS?
                           0C 1F 0171 338      BLSSU 70$            ; EVENT FLAG
                           54 0000*8F AA 0173 339      BICW #IOSM_SETEVF,R4 ; DON'T SET EVENT FLAG ON BUFFER FULLS
                           10 A6 50 D0 0178 340      MOVL R0,IBFSL_COMPLADDR(R6) ; SAVE COMPLETION ROUTINE ADDRESS
                           017C 341
                           017C 342 65$: ; USE DEFAULT EVENT FLAG
                           50 16 9A 017C 343      MOVZBL #DEFEVFLG,RO
                           017F 344
                           017F 345 70$: ; SAVE EVENT FLAG NUMBER IN R0

```

START SWEEP ROUTINES

```

4E A6 50 90 017F 346      MOVB R0,IBFSB_EFN(R6)
                            ; PROCESS DELAY
53 97 0183 347
09 19 0183 348
50 87 00 0187 349
04 13 018A 350
38 A6 60 B0 018C 351
                            DECB R3 : DECR. ARG COUNT
                            BLSS 80$ : ARG OMITTED
                            MOVL (R7)+,R0 : ADDRESS OF DELAY VALUE
                            BEQL 80$ : DEFAULTED
                            MOVW (R0),IBFSL_CMDTBL+CMTSW_DELAY(R6) : STORE IN COMMAND TABLE
                            ; DON'T PROCESS ICHN AND NCHN IF SETADC WAS CALLED
1A 4C A6 03 E0 0190 355 80$: BBS #FLG_V_SETADC,IBFSW_FLAGS(R6),90$ ; BR. IF IT WAS CALLED
                            ; PROCESS ICHN
53 97 0195 356
09 19 0197 357
50 87 00 0199 358
04 13 019C 359
3A A6 60 90 019E 360
                            DECB R3 : DECR. ARG COUNT
                            BLSS 85$ : ARG OMITTED
                            MOVL (R7)+,R0 : GET ADDRESS OF ICHN VALUE
                            BEQL 85$ : DEFAULTED
                            MOVW (R0),IBFSL_CMDTBL+CMTSB_ICHN(R6) : STORE ICHN
                            ; PROCESS NCHN
53 97 01A2 361
09 19 01A4 362
50 87 00 01A6 363
04 13 01A9 364
3C A6 60 B0 01AB 365 85$: BEQL 90$ : DECR. ARG COUNT
                            BLSS 90$ : ARG OMITTED
                            MOVL (R7)+,R0 : GET ADDRESS OF NCHN VALUE
                            BEQL 90$ : DEFAULTED
                            MOVW (R0),IBFSL_CMDTBL+CMTSW_NCHN(R6) : STORE NCHN
                            ; SET CHANNEL SELECTION BITS IN MODE WORD
50 34 A6 D0 01AF 372 90$: MOVL IBFSL_CMDTBL+CMTSL_RCLADDR(R6),R0 ; RCL ADDRESS SUPPLIED?
0F 13 01B3 373
                            BEQL 94$ ; NO
                            ; HAVE RCL ADDRESS. NOTE CHANNEL BITS ARE ALREADY ZERO. FIND RCL LENGTH
51 D4 01B5 374 92$: CLRL R1
51 02 C0 01B7 375 92$: ADDL #2,R1 : ADD 2 TO LENGTH
80 B5 01BA 376 TSTW (R0)+ : AT END OF RCL?
F9 18 01BC 377 92$: BGEO 92$ : NOT YET
30 A6 51 D0 01BE 378 92$: MOVL R1,IBFSL_CMDTBL+CMTSL_RCLLEN(R6) : STORE RCL LENGTH
10 11 01C2 379
                            BRB 100$ :
                            ; IS NCHN = 1?
3C A6 01 B1 01C4 384 94$: CMPW #1,IBFSL_CMDTBL+CMTSW_NCHN(R6)
06 13 01C8 385 BEQL 95$ : YES, SET SINGLE CHANNEL BIT
21 A6 02 88 01CA 386 BISB #2,IBFSL_CMDTBL+CMTSW_MODE+1(R6); NO, SET SEQUENTIAL CHANNEL BIT
04 11 01CE 387 BRB 100$ :
21 A6 01 88 01D0 388 95$: BISB #1,IBFSL_CMDTBL+CMTSW_MODE+1(R6); SET SINGLE CHANNEL BIT
                            ; NUMBER TO APPEND TO LOGICAL NAME
53 D4 01D4 389 100$: CLRL R3 : USED TO ASSIGN CHANNEL TO
01D6 390 100$: MOVL IBFSL_LAMSKB(R6),R0 : GET ADDRESS OF LAMASK BUFFER
                            BEQL 110$ : THERE IS NONE
50 1C A6 D0 01D6 391 100$: MOVW (R0),IBFSL_CMDTBL+CMTSB_STWRDN(R6) : MOVE CONTENTS
08 13 01DA 392 100$: MOVZWL 6(R0),R3 : GET NUMBER TO APPEND TO LOGICAL NAME
                            ; USED TO ASSIGN CHANNEL TO
40 A6 60 7D 01DC 393
53 06 A0 3C 01E0 394
                            01E4 395
                            01E4 396
                            01E4 397
                            01E4 398 110$: ; ASSIGN CHANNEL
52 4A A6 3E 01E4 399 110$: MOVAW IBFSW_CHAN(R6),R2 : ADDRESS OF PLACE TO STORE CHANNEL #
02F4 30 01E8 400 BSBW LPASSASSIGN : ASSIGN CHANNEL
4B 50 E9 01EB 401 BLBC R0,115$ : ERROR
                            01EE 402

```

52	00000000'EF	9E	01EE	403		;	NOW SET UP TO DO QIO			
53	00000000'EF	9E	01EE	404	MOVAB	LPASSCMPLAST,R2	:	ADDRESS OF QIO COMPLETE AST		
55	00000000'EF	9E	01F5	405	MOVAB	LPASSBFRAST,R3	:	ADDRESS OF BUFFER AST		
			01FC	406	MOVAB	LPASSOVRAST,R5	:	ADDRESS OF OVER/UNDERRUN AST		
			0203	407	SQ10_S	IBFSB_EFN(R6),-	:	EVENT FLAG		
			0203	408		IBFSW_CHAN(R6),-	:	CHANNEL		
			0203	409		R4,-	:	I/O FUNCTION CODE		
			0203	410		IBFSQ_IOSB(R6),-	:	I/O STATUS BLOCK		
			0203	411		(R2),=	:	COMPLETION AST ADDRESS		
			0203	412		R6,-	:	AST PARAMETER (ADDRESS OF IBUF ARRAY)		
			0203	413		IBFSL_CMDTBL(R6),-	:	ADDRESS OF COMMAND TABLE		
			0203	414		#40,-	:	LENGTH OF COMMAND TABLE		
			0203	415		R3,-	:	NORMAL BUFFER AST ADDRESS		
			0203	416		R5	:	OVER/UNDERRUN AST ADDRESS		
10	50	E8	0226	417	BLBS	R0,1158	:	SUCCESSFUL QIO		
			0229	418						
			0229	419						
50	DD		0229	420	;	ERROR IN QIO				
			0228	421	PUSHL	R0	:	SAVE STATUS		
50	8ED0		0236	422	SDASSGN_S		IBFSW_CHAN(R6)	:	DEASSIGN CHANNEL	
			0239	423	POPL	R0		:	RESTORE STATUS	
			0239	424						
0A	6C	91	0239	425						
	09	1F	023C	426	1158:	: ALL ERRORS AND SUCCESS COME HERE WITH STATUS IN R0				
51	28	AC	023E	427	CMPB	(AP),#10		IND SPECIFIED?		
			03	13	BLSSU	120\$		NO		
61	50	DO	0242	428	MOVL	40(AP),R1		GET ADDRESS OF IND		
			0244	429	BEQL	120\$		DEFAULTED		
			0247	430	MOVL	R0,(R1)		STORE STATUS		
			04	0247	431	120\$:	RET			

0248 433 .SBTTL LPASSTPSWP - STOP SWEEP
 0248 434 **
 0248 435 FUNCTIONAL DESCRIPTION:
 0248 436 THIS ROUTINE STOPS SWEEPS. A SWEEP CAN BE STOPPED IN TWO WAYS:
 0248 437 EITHER BY SETTING THE STOP BIT IN THE USER STATUS WORD (WHICH
 0248 438 STOPS AT THE END OF THE CURRENT BUFFER) OR BY ISSUING A CANCEL I/O
 0248 439 (WHICH STOPS IT IMMEDIATELY).
 0248 440
 0248 441 CALLING SEQUENCE:
 0248 442
 0248 443
 0248 444 CALLS/G
 0248 445
 0248 446 INPUT PARAMETERS:
 0248 447
 0248 448 IBUF(AP) ADDRESS OF IBUF ARRAY
 0248 449 IWHEN(AP) ADDRESS OF BYTE WHICH SPECIFIES WHEN TO
 0248 450 STOP SWEEP (0 = IMMED. NON-0 = AT END OF BFR.)
 0248 451 IND(AP) ADDRESS OF LONGWORD TO RECEIVE STATUS
 0248 452
 0248 453 IMPLICIT INPUTS:
 0248 454
 0248 455 VARIOUS FIELDS IN IBUF ARRAY
 0248 456
 0248 457 OUTPUT PARAMETERS:
 0248 458
 0248 459 IND(AP) ADDRESS OF LONGWORD TO RECEIVE STATUS
 0248 460
 0248 461 IMPLICIT OUTPUTS:
 0248 462
 0248 463 NONE
 0248 464
 0248 465 COMPLETION CODES:
 0248 466
 0248 467 1 INDICATES SUCCESS
 0248 468 VARIOUS ERRORS RETURNED BY SCANCEL
 0248 469
 0248 470 SIDE EFFECTS:
 0248 471
 0248 472 FIELDS IN THE IBUF ARRAY MAY BE MODIFIED
 0248 473
 0248 474 --
 0040 475
 0248 476 .ENTRY LPASSTPSWP,"M<R6>"
 024A 477
 56 04 AC D0 024A 478 MOVL 4(AP),R6 : GET ADDRESS OF IBUF ARRAY
 024E 479
 02 6C 91 024E 480 CMPB (AP),#2 : IS IWHEN SUPPLIED?
 14 1F 0251 481 BLSSU 20S : NO
 50 08 AC D0 0253 482 MOVL 8(AP),R0 : GET ADDRESS OF IWHEN
 0E 13 0257 483 BEQL 20S : DEFAULTED
 60 D5 0259 484 TSTL (R0) : DETERMINE WHEN
 0A 13 025B 485 BEQL 20S : IMMEDIATELY
 025D 486
 49 A6 40 8F 025D 487 : AT END OF CURRENT BUFFER - SET STOP BIT IN USW
 50 01 D0 025D 488 BISB #^X40,IBFSW_USW+1(R6) :
 0262 489 MOVL #1,R0 : SUCCESS

LPASSTPSWP - STOP SWEEP

16-SEP-1984 01:44:18 VAX/VMS Macro V04-00
5-SEP-1984 01:32:23 [IOSUP.SRC]LASWEEP.MAR;1Page 11
(5)

0B	11	0265	490		BRB	40\$	
		0267	491				
		0267	492	20\$: : ISSUE CANCEL I/O			
		0267	493	\$CANCEL_S	IBFSW_CHAN(R6)	: RETURNS STATUS IN R0	
		0272	494				
03	6C	91	0272	495	CMPB (AP),#3		: IND SUPPLIED?
	09	1F	0275	496	BLSSU 60\$: NO
51	0C	AC	00	0277	497	MOVL 12(AP),R1	: YES, GET ADDRESS
	03	13	0278	498	BEQL 60\$: DEFAULTED
61	50	00	027D	499	MOVL R0,(R1)		: STORE STATUS IN IND
	04	0280	500	60\$: RET			

0281 502 .SBTTL LPASCLOCKA - SET CLOCK A RATE
 0281 503 ++
 0281 504 : FUNCTIONAL DESCRIPTION:
 0281 505 : THIS ROUTINE SETS THE RATE FOR CLOCK A
 0281 506 :
 0281 507 : CALLING SEQUENCE:
 0281 508 :
 0281 509 :
 0281 510 : CALLS/G
 0281 511 :
 0281 512 : INPUT PARAMETERS:
 0281 513 :
 0281 514 : IRATE(AP) ADDRESS OF LONGWORD CONTAINING CLOCK RATE
 0281 515 : IPRSET(AP) ADDRESS OF WORD CONTAINING CLOCK PRESET
 0281 516 : IND(AP) ADDRESS OF LONGWORD TO RECEIVE COMPLETION STATUS
 0281 517 : CHAN(AP) ADDRESS OF WORD CONTAINING NUMBER TO IDENTIFY
 WHICH LPA-11
 0281 518 :
 0281 519 :
 0281 520 : IMPLICIT INPUTS:
 0281 521 :
 0281 522 : NONE
 0281 523 :
 0281 524 : OUTPUT PARAMETERS:
 0281 525 :
 0281 526 : IND(AP) ADDRESS OF LONGWORD TO RECEIVE COMPLETION STATUS
 0281 527 :
 0281 528 : IMPLICIT OUTPUTS:
 0281 529 :
 0281 530 : NONE
 0281 531 :
 0281 532 : COMPLETION CODES:
 0281 533 :
 0281 534 : 1 INDICATES SUCCESS
 0281 535 : VARIOUS VMS SYSTEM STATUS CODES INDICATE ERRORS
 0281 536 :
 0281 537 : SIDE EFFECTS:
 0281 538 :
 0281 539 : NONE
 0281 540 :
 0281 541 :--
 0281 542 :
 00FC 0281 543 :.ENTRY LPASCLOCKA,^M<R2,R3,R4,R5,R6,R7>
 55 01 00 0283 544 :
 0283 545 : MOVL #1,RS ; MODE WORD
 0283 546 :
 0286 547 : BUILD CLOCK STATUS. IF IRATE IS >= 0 THEN USE AS IS. IF
 0286 548 : IRATE IS < 0 THEN SET SCHMITT TRIGGER 1 INTERRUPT ENABLE (BIT 14).
 56 00004141 8F 00 0286 549 : MOVL #^X4141,R6 : CLOCK STATUS INCLUDING BIT 14
 50 04 BC 00 0280 550 : MOVL R4(AP),R0 : PUT IRATE IN R0
 0280 551 : BLSS 10\$: ITS < 0. LEAVE RATE = 0 (BITS 1 - 3)
 56 56 4000 8F AA 0293 552 : BICW #^X4000,R6 : ITS >= 0. CLEAR BIT 14
 03 01 50 F0 0298 553 : INSV R0,#1,\$,R6 : INSERT RATE
 57 08 BC 3C 029D 554 10\$: MOVZWL R8(AP),R7 : CLOCK PRESET
 02A1 555 :
 02A1 556 : PUT # OF ARGS LEFT IN R0, POINTER TO THEM IN R1, AND
 02A1 557 : JOIN COMMON CLOCK ROUTINE
 50 6C 02 83 02A1 558 : SUBBS #2,(AP),R0 : # OF ARGS LEFT

LPASSWEEP
V04-000

LPASCLOCKA - SET CLOCK A RATE

C 11

16-SEP-1984 01:44:18 VAX/VMS Macro V04-00
5-SEP-1984 01:32:23 [IOSUP.SRC]LASWEEP.MAR;1

Page 13
(6)

51 OC AC DE 02A5 559
38 11 02A9 560

MOVAL
BRB

12(AP),R1
CLKCOM

: ADDRESS OF NEXT ARG
: JOIN COMMON ROUTINE

LPA
V04-

LPAS\$CLOCKB - SET CLOCK B RATE

16-SEP-1984 01:46:18 VAX/VMS Macro V04-00
5-SEP-1984 01:32:23 [IOSUP.SRC]LASWEEP.MAR;1

02AB 562 .SBTTL LPAS\$CLOCKB - SET CLOCK B RATE
 02AB 563 :++
 02AB 564 : FUNCTIONAL DESCRIPTION:
 02AB 565 :
 02AB 566 : THIS ROUTINE SETS THE RATE FOR CLOCK B
 02AB 567 :
 02AB 568 : CALLING SEQUENCE:
 02AB 569 :
 02AB 570 : CALLS/G
 02AB 571 :
 02AB 572 : INPUT PARAMETERS:
 02AB 573 :
 02AB 574 : I RATE(AP) ADDRESS OF LONGWORD CONTAINING CLOCK RATE
 02AB 575 : I PRSET(AP) ADDRESS OF WORD CONTAINING CLOCK PRESET
 02AB 576 : MODE(AP) ADDRESS OF A WORD WHICH SPECIFIES OPTIONS
 02AB 577 :
 02AB 578 : BIT 0 SET INDICATES OPERATE CLOCK B IN
 02AB 579 : NON-INTERRUPT MODE
 02AB 580 :
 02AB 581 : IND(AP) ADDRESS OF LONGWORD TO RECEIVE COMPLETION STATUS
 02AB 582 : CHAN(AP) ADDRESS OF WORD CONTAINING NUMBER TO IDENTIFY
 02AB 583 : WHICH LPA-11
 02AB 584 :
 02AB 585 : IMPLICIT INPUTS:
 02AB 586 :
 02AB 587 : NONE
 02AB 588 :
 02AB 589 : OUTPUT PARAMETERS:
 02AB 590 :
 02AB 591 : IND(AP) ADDRESS OF LONGWORD TO RECEIVE COMPLETION STATUS
 02AB 592 :
 02AB 593 : IMPLICIT OUTPUTS:
 02AB 594 :
 02AB 595 : NONE
 02AB 596 :
 02AB 597 : COMPLETION CODES:
 02AB 598 :
 02AB 599 : 1 INDICATES SUCCESS
 02AB 600 : VARIOUS VMS SYSTEM STATUS CODES INDICATE ERRORS
 02AB 601 :
 02AB 602 : SIDE EFFECTS:
 02AB 603 :
 02AB 604 :
 02AB 605 :
 02AB 606 :--
 02AB 607 :
 00FC 608 .ENTRY LPAS\$CLOCKB,"M<R2,R3,R4,R5,R6,R7>
 02AD 609
 56 00000041 55 11 D0 02AD 610 MOVL #^X11,R5 : MODE WORD
 50 04 AC 8F D0 02B0 611 MOVL #^X41,R6 : CLOCK STATUS
 50 1E 13 00 02B7 612 MOVL 4(AP),R0 : ADDRESS OF IRATE
 50 60 D0 02BB 613 BEQL 40\$: DEFAULTED
 50 19 13 02BD 614 MOVL (R0),R0 : GET IRATE
 56 03 01 50 F0 02C0 615 BEQL 40\$: ZERO
 56 04 0C BC E9 02C2 616 INSV R0,#1,#3,R6 : INSERT IRATE
 56 40 8F 8A 02CB 617 BLBC 212(AP),20\$: BR. IF LEAVE INTERRUPTS ENABLED
 56 02CB BICB #^X40,R6 : CLEAR INTERRUPT ENABLE

03	0C	BC	01	F1	02CF	619	20\$:	BBC	#1,012(AP),30\$: BR. IF DON'T SET FEED B TO A
57	56	20	08	88	02D4	620		BISB	#^X20,R6	: SET FEED B TO A
57	08	BC	3C	3C	02D7	621	30\$:	MOVZWL	08(AP\$),R7	: PRESET
50	6C	10	03	83	02DB	622	40\$:	: GET # OF ARGS LEFT IN R0, ADDRESS OF NEXT ARG IN R1		
51	10	AC	DE	DE	02DB	623		\$UBB3	#3,(AP),R0	: NUMBER OF ARGS LEFT
					02DF	624		MOVAL	16(AP),R1	: ADDRESS OF NEXT ARG
					02E3	625				
					02E3	626		CLKCOM:	: BOTH CLOCK ROUTINES COME HERE FOR COMMON PROCESSING	
					02E3	627		CLRQ	R3	: R3 WILL HOLD LPA CHANNEL
					02E5	628				: R4 WILL HOLD ADDR. OF IND
					02E5	629				
					02E5	630				
50	97				02E5	631		DECBL	R0	: DECR. ARG COUNT
OF	19				02E7	632		BLSS	50\$: NO MORE ARGS
54	81	DO	DO	02E9	633		MOVL	(R1)+,R4		: ADDR. OF IND
				02EC	634					
50	97			02EC	635		DECBL	R0	: DECR. ARG COUNT	
08	19			02EE	636		BLSS	50\$: NO MORE ARGS	
53	81	DO	DO	02FO	637		MOVL	(R1)+,R3		: ADDRESS OF LPA CHANNEL
03	13			02F3	638		BEQL	50\$: DEFAULTED
53	63	3C	3C	02F5	639		MOVZWL	(R3),R3		: GET CHANNEL NUMBER
				02F8	640					
				02F8	641	50\$:	: NOW ALLOCATE SPACE ON STACK TO RECEIVE CHANNEL NUMBER ASSIGNED			
				02F8	642		: AND SPACE FOR I/O STATUS BLOCK FOR QIO.			
5E	0C	C2	02F8	643			SUBL	#12,SP		: 3 LONGWORDS SHOULD DO IT
52	SE	DO	02FB	644			MOVL	SP,R2		: PLACE TO STORE CHANNEL NUMBER ASSIGNED
01DE	30	02FE	645				BSBW	LPA\$ASSIGN		: ASSIGN CHANNEL
45	50	E9	0301	646			BLBC	R0,90\$: ERROR
				0304	647					
				0304	648		: NOW SET CLOCK			
				0304	649		\$QIOW_S #CLKEVFLG,-			
				0304	650		(R2),-			
				0304	651		#IOS SETCLOCK,-			
				0304	652		: I/O FUNCTION CODE			
				0304	653		4(R2),-			
				0304	654					
				0304	655		BLBC	R5,R6,R7		
50	13	50	E9	0324	656		MOVZWL	R0,80\$		
04	A2	3C	0327	657			BLBC	4(R2),R0		
OC	50	E9	0328	658				R0,80\$		
				032E	659					
				032E	660		: NOW DEASSIGN CHANNEL			
				032E	661		\$DASSGN_S (R2)			
OF	11	0338	662				BRB	90\$: STATUS IS IN R0
				033A	663	80\$:				
50	DD	033A	664				: ERROR IN QIO - SAVE STATUS BEFORE DEASSIGNING CHANNEL			
				033A	665		PUSHL	R0		: SAVE STATUS
50	8EDC	0346	666				\$DASSGN_S (R2)			
				0346	667		POPL	R0		
				0349	668	90\$:	TSTL	R4		
54	D5	0349	669				BEQL	95\$: IND SPECIFIED?
03	13	0348	670				MOVL	R0,(R4)		: NO
64	50	DO	034D	671	95\$:		RET			: YES, STORE STATUS
				04	0350					

0351 673 .SBTTL LPASLAMSKS - SET MASKS BUFFER
 0351 674 :++
 0351 675 : FUNCTIONAL DESCRIPTION:
 0351 676 :
 0351 677 : THIS ROUTINE PERFORMS TWO COMPLETELY SEPARATE FUNCTIONS.
 0351 678 : FIRST IT MUST BE CALLED BY PROGRAMS THAT UTILIZE TWO OR MORE
 0351 679 : LPA-11'S TO IDENTIFY WHICH LPA-11 IS TO BE USED IN A SUBSEQUENT
 0351 680 : START SWEEP CALL. THE SECOND ARGUMENT IS A NUMBER WHICH IS APPENDED
 0351 681 : TO THE LOGICAL NAME THAT IS USED TO ASSIGN A CHANNEL TO. IT IS
 0351 682 : ASSUMED THAT THE USER HAS ASSIGNED THE RESULTANT LOGICAL NAME
 0351 683 : TO THE APPROPRIATE LPA-11.
 0351 684 : SECONDLY, THIS ROUTINE MUST BE CALLED BY PROGRAMS THAT UTILIZE
 0351 685 : DIGITAL INPUT STARTING OR EVENT MARKING. ARGUMENTS CAN SUPPLIED
 0351 686 : FOR THE DIGITAL INPUT START WORD AND MASK AND FOR THE EVENT MARK
 0351 687 : WORD AND MASK.
 0351 688 :
 0351 689 : CALLING SEQUENCE:
 0351 690 :
 0351 691 : CALLS/G
 0351 692 :
 0351 693 : INPUT PARAMETERS:
 0351 694 :
 0351 695 : LAMSKB(AP) ADDRESS OF 8 BYTE ARRAY
 0351 696 : NUM(AP) ADDRESS OF WORD CONTAINING NUMBER TO APPEND TO LOG. NAME
 0351 697 : IUNIT(AP) UNUSED (PRESENT FOR RSX-11M COMPATIBILITY)
 0351 698 : STWRDN(AP) ADDRESS OF BYTE CONTAINING DIGITAL START CHANNEL
 0351 699 : EVMRKN(AP) ADDRESS OF BYTE CONTAINING EVENT MARK CHANNEL
 0351 700 : STWRDM(AP) ADDRESS OF WORD CONTAINING DIGITAL START MASK
 0351 701 : EVMRKM(AP) ADDRESS OF WORD CONTAINING EVENT MARK MASK
 0351 702 : IND(AP) ADDRESS OF LONGWORD TO RECEIVE STATUS
 0351 703 :
 0351 704 : IMPLICIT INPUTS:
 0351 705 :
 0351 706 : NONE
 0351 707 :
 0351 708 : OUTPUT PARAMETERS:
 0351 709 :
 0351 710 : IND(AP) ADDRESS OF LONGWORD TO RECEIVE STATUS
 0351 711 :
 0351 712 : IMPLICIT OUTPUTS:
 0351 713 :
 0351 714 : THE 8 BYTE ARRAY IS FILLED IN
 0351 715 :
 0351 716 : COMPLETION CODES:
 0351 717 :
 0351 718 : 1 INDICATES SUCCESS (THIS ROUTINE ALWAYS RETURNS SUCCESS.
 THE ARGUMENT IS PRESENT FOR COMPATIBILITY ONLY)
 0351 719 :
 0351 720 :
 0351 721 : SIDE EFFECTS:
 0351 722 :
 0351 723 : NONE
 0351 724 :
 0351 725 :--
 000C 726 .ENTRY LPASLAMSKS,"M<R2,R3>"
 0351 727
 0353 728 MOVAL 4(AP),R3 ; R3 CONTAINS ADDRESS OF ARGUMENT LIST
 0353 729

53 04 AC DE 0353
 000C 726 .ENTRY LPASLAMSKS,"M<R2,R3>"
 0351 727
 0353 728 MOVAL 4(AP),R3 ; R3 CONTAINS ADDRESS OF ARGUMENT LIST
 0353 729

LPASLAMSKS - SET MASKS BUFFER								
52	83	D0	0357	730	MOVL	(R3)+,R2	: R2 CONTAINS ADDRESS OF 8 BYTE ARRAY	
	62	7C	035A	731	CLRR	(R2)	: CLEAR ARRAY	
			035C	732				
			035C	733				
51	6C	02	83	035C	734	: PROCESS NUM		
	4B	19	0360	735	SUBB	#2,(AP),R1	: R1 CONTAINS # OF REMAINING ARGS	
50	83	D0	0362	736	BLSS	90\$: NO MORE ARGS	
	04	13	0365	737	MOVL	(R3)+,R0	: GET ADDRESS OF NUM	
06	A2	60	80	0367	738	BEQL	10\$: DEFAULTED
			0368	739	MOVW	(R0),6(R2)	: STORE IN ARRAY	
			0368	740	10\$:			
51	02	82	0368	741	: PROCESS STWRDN			
	3D	19	036E	742	SUBB	#2,R1	: ENOUGH ARGS?	
	83	D5	0370	743	BLSS	90\$: NOPE	
50	83	D0	0372	744	TSTL	(R3)+	: SKIP OVER UNUSED ARG	
	03	13	0375	745	MOVL	(R3)+,R0	: GET ADDRESS OF STRWDN	
62	60	90	0377	746	BEQL	20\$: DEFAULTED	
			037A	747	MOVB	(R0),(R2)	: STORE IN ARRAY	
			037A	748	20\$:			
51	97	037A	749		: PROCESS EVMRKN			
	2F	19	037C	750	DEC8	R1	: ENOUGH ARGS?	
50	83	D0	037E	751	BLSS	90\$: NOPE	
	04	13	0381	752	MOVL	(R3)+,R0	: GET ADDRESS OF EVMRKN	
01	A2	60	90	0383	753	BEQL	30\$: DEFAULTED
			0387	754	MOVW	(R0),1(R2)	: STORE IN ARRAY	
			0387	755	30\$:			
51	97	0387	756		: PROCESS STWRDM			
	22	19	0389	757	DEC8	R1	: ENOUGH ARGS?	
50	83	D0	038B	758	BLSS	90\$: NOPE	
	04	13	038E	759	MOVL	(R3)+,R0	: GET ADDRESS OF STWRDM	
02	A2	60	80	0390	760	BEQL	40\$: DEFAULTED
			0394	761	MOVW	(R0),2(R2)	: STORE IN ARRAY	
			0394	762	40\$:			
51	97	0394	763		: PROCESS EVMRKM			
	15	19	0396	764	DEC8	R1	: ENOUGH ARGS?	
50	83	D0	0398	765	BLSS	90\$: NOPE	
	04	13	0398	766	MOVL	(R3)+,R0	: GET ADDRESS OF EVMRKN	
04	A2	60	80	039D	767	BEQL	50\$: DEFAULTED
			03A1	768	MOVW	(R0),4(R2)	: STORE IN ARRAY	
			03A1	769	50\$:			
51	97	03A1	770		: PROCESS IND			
	08	19	03A3	771	DEC8	R1	: ENOUGH ARGS?	
50	83	D0	03A5	772	BLSS	90\$: NOPE	
	03	13	03A8	773	MOVL	(R3)+,R0	: GET ADDRESS OF IND	
60	01	D0	03AA	774	BEQL	90\$: DEFAULTED	
			03AD	775	MOVW	#1,(R0)	: STORE SUCCESS STATUS	
			04	03AD	776	90\$:	RET	

03AE 778 .SBTTL LPASSETADC - SET CHANNEL PARAMETERS
 03AE 779 :++
 03AE 780 : FUNCTIONAL DESCRIPTION:
 03AE 781 : THIS ROUTINE SETS THE CHANNEL SAMPLING PARAMETERS.
 03AE 782 :
 03AE 783 : CALLING SEQUENCE:
 03AE 784 : CALLS/G
 03AE 785 :
 03AE 786 : INPUT PARAMETERS:
 03AE 787 :
 03AE 788 :
 03AE 789 :
 03AE 790 : IBUF(AP) ADDRESS OF IBUF ARRAY
 03AE 791 : IFLAG(AP) UNUSED (PRESENT FOR COMPATIBILITY WITH RSX-11M)
 03AE 792 : ICHN(AP) IF INC IS DEFAULTED OR NON-ZERO, THIS IS THE ADDRESS
 03AE 793 : OF A BYTE CONTAINING THE INITIAL CHANNEL NUMBER.
 03AE 794 : IF INC = 0, THIS IS THE ADDRESS OF A RANDOM
 03AE 795 : CHANNEL LIST.
 03AE 796 : NCHN(AP) ADDRESS OF A WORD CONTAINING NUMBER OF SAMPLES TO
 03AE 797 : BE TAKEN PER SAMPLE SEQUENCE.
 03AE 798 : INC(AP) ADDRESS OF A BYTE CONTAINING THE CHANNEL INCREMENT.
 03AE 799 : IF THIS BYTE CONTAINS 0, THEN ICHN IS THE ADDRESS
 03AE 800 : OF A RANDOM CHANNEL LIST.
 03AE 801 : IND(AP) ADDRESS OF A LONGWORD TO RECEIVE STATUS.
 03AE 802 :
 03AE 803 : IMPLICIT INPUTS:
 03AE 804 :
 03AE 805 : NONE
 03AE 806 :
 03AE 807 : OUTPUT PARAMETERS:
 03AE 808 :
 03AE 809 : IND(AP) ADDRESS OF A LONGWORD TO RECEIVE STATUS
 03AE 810 :
 03AE 811 : IMPLICIT OUTPUTS:
 03AE 812 :
 03AE 813 : NONE
 03AE 814 :
 03AE 815 : COMPLETION CODES:
 03AE 816 :
 03AE 817 : 0 INDICATES LPASSETIBF WAS NOT CALLED PRIOR TO THIS CALL
 03AE 818 : 1 INDICATES SUCCESS
 03AE 819 :
 03AE 820 : SIDE EFFECTS:
 03AE 821 :
 03AE 822 : VARIOUS FIELDS IN THE IBUF ARRAY ARE MODIFIED
 03AE 823 :
 03AE 824 :--
 005C 03AE 825 .ENTRY LPASSETADC,"M<R2,R3,R4,R6>
 03AE 826 :
 03AE 827 :
 56 04 50 D4 03B0 828 CLRL R0 : STATUS
 1234 8F AC D0 03B2 829 MOVL 4(AP),R6 : ADDRESS OF IBUF ARRAY
 66 B1 03B6 830 CMPW IBFSQ_IOST(R6),#INITCODE : VERIFY LPASSETIBF WAS CALLED
 3D 12 03B8 831 BNEQ 808 : IT WASN'T - ERROR
 53 0C AC DE 03B0 832 MOVAL 12(AP),R3 : POINT TO ICHN ARG
 03C1 833 :
 03C1 834 ; PROCESS ICHN

52 6C 03 83 03C1 835 SUBB3 #3 (AP),R2 ; ENOUGH ARGS?
 54 2D 19 03C5 836 BLSS 70\$; NOPE
 54 83 00 03C7 837 MOVL (R3)+,R4 ; R4 = ADDRESS OF ICHN OR RCL
 52 97 03CA 839 ; PROCESS NCHN
 51 1E 19 03CC 840 DECB R2 ; ENOUGH ARGS?
 51 83 00 03CE 841 BLSS 20\$; NOPE
 51 04 13 03D1 842 MOVL (R3)+,R1 ; GET ADDRESS OF NCHN
 3C A6 61 B0 03D3 843 BEQL 10\$; DEFAULTED
 3C A6 61 B0 03D7 844 MOVW (R1),IBFSL_CMDTBL+CMTSW_NCHN(R6) ; STORE NCHN
 52 97 03D7 846 10\$: ; PROCESS INC
 51 11 19 03D9 847 DECB R2 ; ENOUGH ARGS?
 51 83 00 03DB 848 BLSS 20\$; NOPE
 51 0C 13 03DE 849 MOVL (R3)+,R1 ; GET ADDRESS OF INC
 3B A6 61 90 03E0 850 BEQL 20\$; DEFAULTED
 3B A6 61 12 03E4 851 MOVW (R1),IBFSL_CMDTBL+CMTSB_INC(R6) ; STORE INC
 3B A6 06 12 03E6 852 BNEQ 20\$; NON-ZERO, SO ICHN IS NOT RCL ADDR.
 34 A6 54 00 03E6 854 ; INC = 0 SO ICHN IS RCL ADDRESS (IN R6)
 34 A6 08 11 03EA 855 MOVL R4,IBFSL_CMDTBL+CMTSL_RCLADDR(R6) ; STORE RCL ADDRESS
 34 A6 08 11 03EC 856 BRB 70\$
 54 D5 03EC 858 20\$: ; INC WAS EITHER DEFAULTED OR NON-ZERO, SO R4 POINTS TO ICHN VALUE
 54 04 13 03EE 859 TSTL R4 ; WAS ICHN DEFAULTED?
 3A A6 64 90 03F0 860 BEQL 70\$; YES
 3A A6 64 90 03F4 861 MOVW (R4),IBFSL_CMDTBL+CMTSB_ICHN(R6) ; NO, STORE ICHN VALUE
 4C A6 08 A8 03F4 863 70\$: ; SUCCESS RETURN
 4C A6 50 D6 03F8 864 BISW #FLG_M_SETADC,IBFSW_FLAGS(R6) ; SET SETADC CALLED BIT
 4C A6 50 D6 03FA 865 INCL R0 ; SET SUCCESS CODE
 06 6C 91 03FA 867 80\$: ; STORE COMPLETION CODE IN IND
 06 09 1F 03FD 868 CMPB (AP),#6 ; ENOUGH ARGS?
 51 18 AC 00 03FF 869 BLSSU 90\$; NO
 51 03 13 0403 870 MOVL 24(AP),R1 ; GET ADDRESS OF IND
 61 50 D0 0405 871 BEQL 90\$; DEFAULTED
 61 50 D0 0408 872 MOVL R0,(R1) ; STORE VALUE IN IND
 61 04 0408 873 RET

Mod ---
 CON ---
 UNS ---
 DEL ---
 BUF ---
 BRO ---
 RES ---
 SCH ---
 EXE ---
 QUE ---
 REC ---
 CHE ---
 GET ---
 SND ---
 BAT ---
 SYM ---
 ACC ---
 RES ---
 ASY ---
 INI ---
 SYS ---
 MAT ---
 LIB ---
 SYS ---

LPASCVADF - CONVERT A/D TO FLOATING POINT

.SBTTL LPASCVADF - CONVERT A/D TO FLOATING POINT
.SBTTL LPASFLT16 - CONVERT UNSIGNED WORD TO FLOATING POINT

0409 875 :++
0409 876 : FUNCTIONAL DESCRIPTION:
0409 877 :
0409 878 : LPASCVADF CONVERTS A NUMBER RETURNED BY AN A/D CONVERTER TO
0409 879 : FLOATING POINT. THE NUMBER IS A SIGNED 12 BIT NUMBER WITH BIT 11
0409 880 : THE SIGN BIT. IT IS FOR THIS REASON THAT ^X0800 IS SUBTRACTED
0409 881 : FROM THE NUMBER BEFORE IT IS CONVERTED TO FLOATING POINT. NOTE THAT
0409 882 : THE ORIGINAL CVADF ROUTINE (ON RSX-11M?) EXPECTED AN A/D VALUE
0409 883 : IN BITS 0 - 11 OF THE WORD AND A GAIN IN BITS 12 - 15. HOWEVER,
0409 884 : THE LPA-11 ON VAX DOES NOT SUPPORT ANY A/D'S THAT SUPPLY A GAIN
0409 885 : IN BITS 12 - 15. THEREFORE, THIS ROUTINE DOES NOT USE THOSE BITS
0409 886 : AS A GAIN.
0409 887 : LPASFLT16 CONVERTS AN UNSIGNED(!) WORD TO FLOATING POINT AND IS
0409 888 : INCLUDED HERE FOR COMPATIBILITY REASONS.

CALLING SEQUENCE:

0409 892 : CALLS/CALLG
0409 893 : THESE ROUTINES MAY BE CALLED AS FUNCTIONS

INPUT PARAMETERS:

0409 894 : IVAL(AP)	ADDRESS OF WORD TO CONVERT
0409 895 : VAL(AP)	ADDRESS OF LONGWORD TO RECEIVE RESULT

IMPLICIT INPUTS:

0409 904 : NONE

OUTPUT PARAMETERS:

0409 908 : VAL(AP)	ADDRESS OF LONGWORD TO RECEIVE RESULT
--------------------	---------------------------------------

IMPLICIT OUTPUTS:

0409 912 : NONE

COMPLETION CODES:

0409 916 : NONE

SIDE EFFECTS:

0409 919 : NONE

0409 920 :--

0409 921 :--

0409 922 :--

0409 923 :--

0409 924 : ENTRY LPASFLT16, "M<>"	
0409 925 : MOVZWL @4(AP),R0	: CONVERT INPUT WORD TO LONGWORD
0409 926 : CVTLF R0,R0	: CONVERT TO FLOATING POINT
0409 927 : BRB CVCOM	
0409 928 : ENTRY LPASCVADF, "M<>"	
0409 929 : SUBW3 #^X0800,@4(AP),R0	: SUBTRACT ^X0800 FROM INPUT
0409 930 : CVTWF R0,R0	: CONVERT TO FLOATING

50 04 BC 0000	0409 924 : ENTRY LPASFLT16, "M<>"
50 50 4E 0000	0409 925 : MOVZWL @4(AP),R0
OC 11 0412	0409 926 : CVTLF R0,R0
	0409 927 : BRB CVCOM
50 04 BC 0800	0409 928 : ENTRY LPASCVADF, "M<>"
50 50 4D 0414	0409 929 : SUBW3 #^X0800,@4(AP),R0
	0409 930 : CVTWF R0,R0

		0420	932		
		0420	933		
		0420	934	CVCOM: : OPTIONAL STORE RESULT	
02	6C	91	0420	935 CMPB (AP),#2	: ENOUGH ARGS SUPPLIED?
	09	1F	0423	936 BLSSU 90\$: NO
51	08	AC	00	0425 937 MOVL 8(AP),R1	: GET ADDRESS OF ARG
	03	13	0429	938 BEQL 90\$: DEFAULTED
61	50	00	0428	939 MOVL R0,(R1)	: STORE RESULT
	04	042E	940 90\$: RET		

Pse

COM

DAT

COD

LI

042F 942 .SBTTL LPASXRATE - COMPUTE CLOCK RATE AND PRESET
 042F 943 ++
 042F 944 FUNCTIONAL DESCRIPTION:
 042F 945
 042F 946 THIS ROUTINE COMPUTES A CLOCK RATE AND PRESET GIVEN A DESIRED
 042F 947 DWELL (INTER-SAMPLE INTERVAL). THE CLOCK RATE IS ALWAYS
 042F 948 THE HIGHEST RATE WHICH WILL PERMIT THE REQUESTED DWELL IN ORDER TO
 042F 949 ACHIEVE THE FINEST RESOLUTION. THIS MAY BE DIFFERENT FOR CLOCK A
 042F 950 AND CLOCK B AS THEY HAVE DIFFERENT MAXIMUM PRESETS. THE CLOCK RATE
 042F 951 IS RETURNED AS A NUMBER (1 - 5) WHICH CAN THEN BE USED AS THE CLOCK
 042F 952 RATE FOR LPASCLOCKA OR LPASCLOCKB. IF CALLED AS A FUNCTION, THE
 042F 953 FUNCTION VALUE IS THE ACTUAL DWELL BEING SUPPLIED, WHICH MAY DIFFER
 042F 954 FROM THE REQUESTED DWELL DUE TO TRUNCATION ERROR.
 042F 955
 042F 956 CALLING SEQUENCE:
 042F 957
 042F 958 CALLS/G
 042F 959 MAY BE CALLED AS A FUNCTION
 042F 960
 042F 961 INPUT PARAMETERS:
 042F 962
 042F 963 DWELL(AP) ADDRESS OF LONGWORD CONTAINING DWELL AS
 042F 964 A FLOATING POINT NUMBER
 042F 965 IRATE(AP) ADDRESS OF A LONGWORD TO RECEIVE THE CLOCK
 042F 966 RATE (1 - 5) (0 INDICATES ERROR)
 042F 967
 042F 968 IPRSET(AP) ADDRESS OF A WORD TO RECEIVE CLOCK PRESET
 042F 969 IFLAG(AP) ADDRESS OF A BYTE WHICH INDICATES WHETHER
 042F 970 THE COMPUTATION IS FOR CLOCK A (MAXIMUM
 042F 971 PRESET = 65535) OR CLOCK B (MAXIMUM PRESET =
 042F 972 255) 0 = CLOCK A. NON-0 = CLOCK B.
 042F 973 IMPLICIT INPUTS:
 042F 974
 042F 975 NONE
 042F 976
 042F 977 OUTPUT PARAMETERS:
 042F 978
 042F 979 IRATE(AP) SEE ABOVE
 042F 980 IPRSET(AP) SEE ABOVE
 042F 981 R0 FUNCTION VALUE. ACTUAL DWELL COMPUTED
 042F 982 AS A FLOATING POINT NUMBER.
 042F 983
 042F 984 NOTE THAT IF THE DESIRED DWELL IS TOO SMALL OR TOO LARGE TO BE
 042F 985 ACHIEVED, THEN BOTH IRATE(AP) AND R0 WILL CONTAIN ZERO
 042F 986
 042F 987 IMPLICIT OUTPUTS:
 042F 988
 042F 989 NONE
 042F 990
 042F 991 COMPLETION CODES:
 042F 992
 042F 993 IF IRATE(AP) OR R0 EQUALS ZERO, THEN THE DESIRED DWELL COULD
 042F 994 NOT BE ACHIEVED.
 042F 995
 042F 996 SIDE EFFECTS:
 042F 997
 042F 998 NONE

LPASXRATE - COMPUTE CLOCK RATE AND PRESE

```

042F 999 :--  

042F 1000 :--  

042F 1001 :--  

003C 042F 1002 :--  

0431 1003 :--  

0431 1004 :--  

0431 1005 :--  

0438 1006 :--  

043B 1007 :--  

043D 1008 :--  

0444 1009 :--  

0444 1010 10$:  

0451 1011 :--  

0451 1012 :--  

0451 1013 :--  

0458 1014 :--  

0458 1015 :--  

0458 1016 20$:  

0458 1017 :--  

045F 1018 :--  

0462 1019 :--  

0464 1020 :--  

0467 1021 :--  

0468 1022 :--  

0468 1023 :--  

0468 1024 :--  

046B 1025 :--  

046D 1026 :--  

046D 1027 30$:  

046D 1028 :--  

0470 1029 :--  

0473 1030 :--  

0476 1031 :--  

0476 1032 :--  

0476 1033 :--  

047A 1034 :--  

047A 1035 :--  

047A 1036 :--  

047A 1037 :--  

047A 1038 :--  

047D 1039 :--  

0481 1040 :--  

0483 1041 :--  

0483 1042 40$:  

0483 1043 :--  

0483 1044 :--  

0485 1045 :--  

0487 1046 :--  

0487 1047 50$:  

0487 1048 :--  

048B 1049 :--
```

.ENTRY LPASXRATE,^M<R2,R3,R4,R5>

: GET MAXIMUM PRESET
MOVF #^F65535.0,R5
TSTB #16(AP)
BEQL 10\$
MOVF #^F255.0,R5

: MAXIMUM PRESET FOR CLOCK A
TEST FLAG
COMPUTATION IS FOR CLOCK A
MAXIMUM PRESET FOR CLOCK B

: DESIRED DWELL
CMPF R4 #^F0.000001
BLSS 40\$
MOVF #^F1000000.0,R3
MOVL #1,R2

: IS IT WITHIN RANGE?
NO
MAXIMUM CLOCK RATE (1 MHZ)
LOOP COUNTER AND CLOCK RATE NUMBER

: CALCULATE PRESET: PRESET = RATE X DWELL
MULF3 R4,R3,R1
CMPF R1,R5
BLEQ 30\$
DIVF #^F10.0,R3
AOBLEQ #5,R2,20\$

: CALCULATED PRESET IN R1
LESS THAN MAXIMUM PRESET ALLOWED?
YES - USE IT
NO - DIVIDE CLOCK RATE BY 10
AND TRY NEXT CLOCK RATE

: IF WE FALL THROUGH THAN DESIRED DWELL IS TOO GREAT TO BE
ACHIEVED WITH THE SLOWEST CLOCK RATE AND LARGEST PRESET.
BRB 40\$

: HAVE CALCULATED PRESET IN R1. ADD 0.5 AND TRUNCATE TO AN INTEGER
ADDF #^F0.5,R1
CVTFL R1,R1
MOVZWL R1,R1

: TO ROUND
CVT TO A LONGWORD TO AVOID OVERFLOW
NOW TRUNCATE TO A WORD

: STORE PRESET (AS TWO'S COMPLEMENT OF CALC. PRESET BECAUSE THAT'S
WHAT THE CLOCKS USE)
MNEGW R1,012(AP)

: NOW CONVERT EVERYTHING BACK TO FLOATING POINT TO GIVE CALLER
ACTUAL DWELL COMPUTED (WHICH MAY BE DIFFERENT DUE TO TRUNCATION ERROR)
CVTLF R1,R1
DIVF3 R3,R1,R0
BRB 50\$

: CONVERT TO FLOATING POINT
ACTUAL DWELL = PRESET / RATE

: ERROR - EITHER DESIRED DWELL WAS TOO GREAT (> 655.35 FOR CLOCK A
OR > 2.55 FOR CLOCK B) OR WAS TOO SMALL (< 0.000001 FOR EITHER CLOCK)
CLRL R2
CLRL R0

: RETURN 0 FOR CLOCK RATE NUMBER
AND 0 FOR COMPUTED DWELL

: STORE CLOCK RATE NUMBER (IN R2)
MOVL R2,08(AP)
RET

: COMPUTED DWELL IN R0 FOR FUNCTION CALL

048C 1051 .SBTTL LPASLOADMC - LOAD MICROCODE
 048C 1052 :++
 048C 1053 : FUNCTIONAL DESCRIPTION:
 048C 1054 :
 048C 1055 : THIS ROUTINE SENDS A REQUEST TO THE LPA-11 MICROCODE LOADER
 048C 1056 : PROCESS TO LOAD A SPECIFIED VERSION OF MICROCODE INTO A SPECIFIED
 048C 1057 : LPA-11. THE LPA-11 IS SPECIFIED BY A NUMBER WHICH IS APPENDED
 048C 1058 : TO A LOGICAL NAME (SEE LPASSASSIGN ROUTINE)
 048C 1059 :
 048C 1060 : CALLING SEQUENCE:
 048C 1061 :
 048C 1062 : CALLS/G
 048C 1063 :
 048C 1064 : INPUT PARAMETERS:
 048C 1065 :
 048C 1066 : ITYPE(AP) ADDRESS OF BYTE CONTAINING TYPE OF
 048C 1067 : MICROCODE TO LOAD
 048C 1068 : 1 = MULTIREQUEST
 048C 1069 : 2 = DED. A/D
 048C 1070 : 3 = DED. D/A
 048C 1071 : INUM(AP) ADDRESS OF WORD CONTAINING NUMBER TO APPEND
 048C 1072 : TO LOGICAL NAME (TO ASSIGN CHANNEL TO)
 048C 1073 : ISTAT(AP) ADDRESS OF LONGWORD TO RECEIVE COMPLETION CODE
 048C 1074 : IERROR(AP) ADDRESS OF LONGWORD TO RECEIVE SECOND
 048C 1075 : LONGWORD OF I/O STATUS BLOCK IF COMPLETION
 048C 1076 : CODE IS SSS_DEVCMDERR, SSS_DEVREQERR, OR
 048C 1077 : SSS_CTRLERR
 048C 1078 :
 048C 1079 : IMPLICIT INPUTS:
 048C 1080 :
 048C 1081 : NONE
 048C 1082 :
 048C 1083 : OUTPUT PARAMETERS:
 048C 1084 :
 048C 1085 : ISTAT(AP) SEE ABOVE
 048C 1086 : IERROR(AP) SEE ABOVE
 048C 1087 :
 048C 1088 : IMPLICIT OUTPUTS:
 048C 1089 :
 048C 1090 : NONE
 048C 1091 :
 048C 1092 : COMPLETION CODES:
 048C 1093 :
 048C 1094 : VARIOUS SYSTEM STATUS'S
 048C 1095 :
 048C 1096 : SIDE EFFECTS:
 048C 1097 :
 048C 1098 : NONE
 048C 1099 :--
 048C 1100 :
 003C 048C 1101 .ENTRY LPASLOADMC,"M<R2,R3,R4,R5>
 048E 1102 :
 54 6C 9A 048E 1103 : GET NUMBER OF ARGUMENTS
 53 D4 0491 1104 : DEFAULT NUMBER TO APPEND TO LOG. NAME
 55 01 DD 0493 1105 :
 0496 1106 :
 0496 1107 : DEFAULT MICROCODE TYPE
 : PROCESS ITYPE

 Sym

 DEL
 DEL
 DEL
 DEQ
 ENQ
 ENT
 EXE
 FET
 FET
 FIN
 FLU
 FMG
 GET
 INI
 INI
 JOB
 JOB
 LIB
 LIB
 LOC
 LOC
 NLA
 OPA
 OPE
 PAL
 PAL

LPASLOADMC - LOAD MICROCODE

```

50 04 54 D7 0496 1108      DECL R4          ; ENOUGH ARGUMENTS SUPPLIED?
      16 19 0498 1109      BLSS 50$          ; NO
      AC D0 049A 1110      MOVL 4(AP),R0    ; GET ADDRESS OF ITYPE
      03 13 049E 1111      BEQL 40$          ; DEFAULTED
      55 60 9A 04A0 1112      MOVZBL (R0),R5 ; GET ITYPE
      60 04A3 1113
      04A3 1114      40$:   ; PROCESS INUM
      54 D7 04A3 1115      DECL R4          ; ENOUGH ARGUMENTS SUPPLIED?
      09 19 04A5 1116      BLSS 50$          ; NO
      AC D0 04A7 1117      MOVL 8(AP),R0    ; GET ADDRESS OF INUM
      03 13 04AB 1118      BEQL 50$          ; DEFAULTED
      53 60 3C 04AD 1119      MOVZWL (R0),R3 ; GET INUM
      04B0 1120
      04B0 1121      50$:   ; ASSIGN CHANNEL TO LPA-11
      52 7E 3E 04B0 1122      MOVAW -(SP),R2 ; GET ADDRESS OF WORD ON TOP OF
      04B3 1123
      0029 30 04B3 1124      BSBW LPASS$ASSIGN ; STACK TO RECEIVE CHANNEL NUMBER
      OB 50 E9 04B6 1125      BLBC R0,70$ ; ASSIGN CHANNEL
      04B9 1126
      04B9 1127      ; NOW SEND REQUEST TO LOADER PROCESS
      55 DD 04B9 1128      PUSHL R5          ; PUSH MICROCODE TYPE
      62 3F 04BB 1129      PUSHAW (R2)     ; PUSH ADDRESS OF CHANNEL
      00000000'EF 02 FB 04BD 1130      CALLS #2,LPASS$NDLDQ ; SEND LOAD REQUEST
      04C4 1131
      04C4 1132      70$:   ; PROCESS ISTAT
      54 D7 04C4 1133      DECL R4          ; ENOUGH ARGUMENTS SUPPLIED?
      16 19 04C6 1134      BLSS 90$          ; NO
      52 0C AC D0 04C8 1135      MOVL 12(AP),R2 ; GET ADDRESS OF ISTAT
      03 13 04CC 1136      BEQL 80$          ; DEFAULTED
      62 50 D0 04CE 1137      MOVL R0,(R2) ; STORE ISTAT
      04D1 1138
      04D1 1139      80$:   ; PROCESS IERROR
      54 D7 04D1 1140      DECL R4          ; ENOUGH ARGUMENTS SUPPLIED?
      09 19 04D3 1141      BLSS 90$          ; NO
      52 10 AC D0 04D5 1142      MOVL 16(AP),R2 ; GET ADDRESS OF IERROR
      03 13 04D9 1143      BEQL 90$          ; DEFAULTED
      62 51 D0 04DB 1144      MOVL R1,(R2) ; STORE IERROR
      04DE 1145
      04DE 1146      90$:   RET
  
```

Symt

PROC
PROC
QUEL
READ

RELE

REMC
REQL
RESE
RESE
RESL
RESL
REW

SCAN

SCH1
SCH1
SCH1
SCH1
SCHE
SCSI
SEAF
SEND
SEND

SGNS
SIGN

SJC
SND
SND
STAF
STAF
STAI
STOI
STOI
STOI
STOI
SYME
SYME
SYSI

				.SBTTL LPASSASSIGN - ASSIGN A CHANNEL TO AN LPA-11
				++ FUNCTIONAL DESCRIPTION:
				THIS ROUTINE ASSIGNS A CHANNEL TO A LOGICAL NAME OF THE FORM LPA11\$ <i>n</i> , WHERE <i>n</i> IS THE VALUE OF ONE OF THE INPUT ARGUMENTS. IT IS ASSUMED THAT THE USER HAS ALREADY ASSIGNED THAT LOGICAL NAME TO AN LPA-11.
				CALLING SEQUENCE:
				BSBW/B
				INPUT PARAMETERS:
				R2 IS THE ADDRESS OF THE LOCATION TO STORE THE CHANNEL NUMBER ASSIGNED R3 CONTAINS THE NUMBER TO BE APPENDED TO THE LOGICAL NAME TO ASSIGN A CHANNEL TO.
				IMPLICIT INPUTS:
				NONE
				OUTPUT PARAMETERS:
				R0 CONTAINS A COMPLETION CODE
				IMPLICIT OUTPUTS:
				NONE
				COMPLETION CODES:
				THE SAME ONES THAT ARE SUPPLIED BY THE \$ASSIGN SYSTEM SERVICE
				SIDE EFFECTS:
				R1 IS NOT PRESERVED
				--
				LPASSASSIGN:
			3C BB	PUSHR #^M<R2,R3,R4,R5> : SAVE SOME REGISTERS
			55 SE DO	MOVL SP,R5 : SAVE STACK POINTER
				04E1 1192
				04E4 1194
				04E4 1195 : CONVERT NUMBER TO ASCII STRING ON STACK
				04E4 1196
				04E4 1197 10\$: : CONVERT NEXT DIGIT
				CLRL R4 : HIGH BITS OF DIVIDEND
			54 53 54	EDIV #10,R3,R3,R4 : QUO.-> R3 REM. -> R4
			0A D6	B1SB3 #^X30,R4,-(SP) : CONVERT TO ASCII AND PUSH ON STACK
			78 04E6	TSTL R3 : REPEAT?
			30 89	BNEQ 10\$: BR. IF YES
			D5 04EF	
			F1 12	
			04F1 1202	
			04F3 1203	
			04F3 1204	: NOW PUSH PREFIX STRING ONTO STACK

LPASSASSIGN - ASSIGN A CHANNEL TO AN LPA

7E	FB05	53 05	D0 04F3	1205		MOVL #DNPREFIXS-1,R3	: LENGTH OF STRING	Sym
		F7 53	90 04F6	1206	20\$:	MOVB DN PREFIX[R3],-(SP)	: PUSH NEXT CHAR ON STACK	---
			F4 04FC	1207		SOBGEQ R3,20\$: REPEAT	SYS
				04FF	1208			SYS
					1209			SYS
53	55	5E	C3 04FF	1210		: NOW BUILD A STRING DESCRIPTOR ON STACK		SYS
		6E	9F 0503	1211		SUBL3 SP,R5,R3	: OVERALL LENGTH OF STRING	SYS
		53	DD 0505	1212		PUSHAB (SP)	: PUSH ADDRESS OF STRING	SYS
		54	5E	D0 0507	1213	PUSHL R3	: PUSH LENGTH	SYS
				050A	1214	MOVL SP,R4	: R4 POINTS TO STRING DESCRIPTOR	UNL
				050A	1215			UNS
				050A	1216			UPD
				0517	1217			UPD
5E	55	D0	0517	1218		MOVL R5,SP	: RESTORE STACK POINTER	VAL
	3C	BA	051A	1219		POPR #^M<R2,R3,R4,R5>	: RESTORE REGISTERS	WAR
		05	051C	1220		RSB	: RETURN CODE IN R0 FROM SASSIGN	WRI
			051D	1221				WRI
			051D	1222				
			051D	1223				
			051D	1224		.END		

LPASSWEEP
Symbol table

E 12

16-SEP-1984 01:44:18 VAX/VMS Macro V04-00
5-SEP-1984 01:32:23 [10SUP.SRC]LASWEEP.MAR;1Page 28
(13)

\$ST1	=	00000001					
BFRVRN	=	000000A3					
CLKCOM	=	000002E3 R	02				
CLKEVFLG	=	00000017					
CMTSB_EVMRKN	=	00000021					
CMTSB_ICHN	=	0000001A					
CMTSB_INC	=	0000001B					
CMTSB_STWRDN	=	00000020					
CMTSB_VBFRMASK	=	00000002					
CMTSL_BFRADDR	=	0000000C					
CMTSL_BFRLLEN	=	00000008					
CMTSL_RCLADDR	=	00000014					
CMTSL_RCLLEN	=	00000010					
CMTSL_USWADDR	=	00000004					
CMTSW_DELAY	=	00000018					
CMTSW_DWELL	=	0000001E					
CMTSW_EVMRKM	=	00000024					
CMTSW_MODE	=	00000000					
CMTSW_NCHN	=	0000001C					
CMTSW_STWRDM	=	00000022					
CVCOM	=	00000420 R	02				
DEFEVFLG	=	00000016					
DNPREFIX	=	00000000 R	02				
DNPREFIXS	=	00000006					
FLG_M_BFRORLSD	=	00000010					
FLG_M_CNTBFRS	=	00000004					
FLG_M_SETADC	=	00000008					
FLG_V_SETADC	=	00000003					
FLG_V_USWSET	=	00000000					
IBFSB_EFN	=	0000004E					
IBFSK_LENGTH	=	000000A8					
IBFSL_CMDTBL	=	00000020					
IBFSL_COMPLADDR	=	00000010					
IBFSL_DEVQBL	=	0000005C					
IBFSL_DEVQFL	=	00000058					
IBFSL_INUQBL	=	00000064					
IBFSL_INUQFL	=	00000060					
IBFSL_LAMSKB	=	0000001C					
IBFSL_LBUF	=	00000014					
IBFSL_NBUF	=	00000018					
IBFSLUSRQBL	=	00000054					
IBFSLUSRQFL	=	00000050					
IBFSQ_BFRLNKS	=	00000068					
IBFSQ_IOSB	=	00000008					
IBFSQ_IOST	=	00000000					
IBFSW_CHAN	=	0000004A					
IBFSW_FLAGS	=	0000004C					
IBFSW_USW	=	00000048					
INITCODE	=	00001234					
IOSM_SETEVF	*****	X	02				
IOS_SETCLOCK	*****	X	02				
IOS_STARTDATA	*****	X	02				
LPASSASSIGN	000004DF R		02				
LPASSBFRAST	*****	X	02				
LPASSCMPLAST	*****	X	02				
LPASSOVRAST	*****	X	02				
LPASSSNLDLDRQ	*****	X	02				

\$2

PSECT name

	Allocation	PSECT No.	Attributes
ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	000000A8 (168.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
_LPA\$CODE	0000051D (1309.)	02 (2.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC BYTE

```
+-----+
! Performance indicators !
+-----+
```

Phase

	Page faults	CPU Time	Elapsed Time
Initialization	16	00:00:00.15	00:00:00.88
Command processing	140	00:00:00.67	00:00:02.35
Pass 1	144	00:00:04.74	00:00:11.42
Symbol table sort	0	00:00:00.20	00:00:00.20
Pass 2	198	00:00:02.55	00:00:04.19
Symbol table output	6	00:00:00.06	00:00:00.06
Psect synopsis output	2	00:00:00.03	00:00:00.15
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	506	00:00:08.41	00:00:19.26

The working set limit was 1500 pages.

28069 bytes (55 pages) of virtual memory were used to buffer the intermediate code.

There were 10 pages of symbol table space allocated to hold 123 non-local and 64 local symbols.

1353 source lines were read in Pass 1, producing 53 object records in Pass 2.

20 pages of virtual memory were used to define 19 macros.

```
+-----+
! Macro library statistics !
+-----+
```

Macro library name

\$255\$DUA28:[SYSLIB]STARLET.MLB;2

Macros defined

14

195 GETS were required to define 14 macros.

There were no errors, warnings or information messages.

MACRO/DISABLE=TRACE/LIS=LIS\$:\$LASWEEP/OBJ=OBJ\$:\$LASWEEP MSRC\$:\$LADEF/UPDATE=(ENH\$:\$LADEF)+MSRC\$:\$LASWEEP/UPDATE=(ENH\$:\$LASWEEP)

0190 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

